**C-lab-2**

**Лабораторная работа №2 (Операторы и выражения-1)**

**Задача №1**

Написать программу, имитирующую работу высотомера бомбы. Бомба

падает с высоты H, которая задается пользователем. В любой

момент времени можно узнать пройденное расстояние по формуле

L = gt^2/2, где g = 9.81 m/c

Высотомер бомбы срабатывает раз в секунду и выводит на терми-

нал текущее значение высоты над поверхностью земли h.

**Пояснение**

В этой задаче мы имитируем сообщения, пересылаемые аппаратурой бомбы на дисплей оператора. Программа должна имитировать секундную задержку между отсчетами и выводить на экран таблицу:

t=00 c h=5000.0 м

t=01 c h=4995.4 м

...

t=31с h=0234.8 м

BABAH!!!

**Состав**

Программа должна состоять из двух функций:

* **float height(int currTime,int startHeight)** - расчет высоты над землей в секунду **currTime** (**startHeight** - начальная высота).
* **main()** - организация диалога.

Как и в задачах первого практикума создаются три файла: **task1.h,task1.c,main1.c**.

**Задача №2**

Написать программу ”Угадай число”. Программа задумывает число в диапа-

зоне от 1 до 100 и пользователь должен угадать его за наименьшее количество

попыток.

**Пояснение**

Пользователь вводит число, а программа подсказывает ему: ”больше”, ”меньше”, ”угадал!”.

**Состав**

Программа должна состоять из двух функций:- **int turn(int value, int secret)** - проверка числа, введенного пользователем. Функция возвращает положительное число, если число пользователя больше задуманного, отрицательное, если меньше и 0 - если числа совпадают. **value** - число пользователя, **secret** - задуманное компьютером число.

* **main()** - организация диалога.

**Задача №3**

Написать программу, выводящую на экран треугольник из звёздочек

**Пояснение**

Треугольник должен выглядеть так:

\*

\*\*\*

\*\*\*\*\*

Количество строк задаётся пользователем с клавиатуры.

**Состав**

Программа должна состоять из двух функций:

* **char \* layout(char buf[],int line,int count)** - формирование одной строки треугольника (**line** - порядковый номер строки с вершины треугольника, начальное значение 1,**count** - общее количество строк). При этом часть строки заполнена пробелами (впереди), а часть символами "звездочка".
* **main()** - организация диалога.

**Задача №4**

Написать программу, переставляющую символы в массиве согласно правилу:

сначала идут латинские буквы, потом цифры. Строка задается в коде програм-

мы в виде случайной последовательности букв и цифр. Пользоваться дополни-

тельными массивами нельзя.

**Пояснение**

Сортировка в данной задаче неприменима ввиду ее трудоемкости. Нужно использовать группировку элементов массива. Программа движется по строке с двух концов к середине и при встрече с нежелательными символами выполняет обмен с символом на другой стороне.

**Состав**

Программа должна состоять из двух функций:

* char\* process(char\* line) - обработка строки.
* **main()** - организация диалога.

**Задача №5**

Написать программу, которая выводит на экран 10 паролей, сгенерированных

случайным образом из латинских букв и цифр, причём буквы должны быть

как в нижнем, так и в верхнем регистрах. Длина пароля - 8 символов.

**Пояснение**

Пример сгенерированного пароля:**Nh1ku83k**

**Состав**

Программа должна состоять из двух функций:

* char \* password(char \* line) - генерация пароля в **line**.
* **main()** - организация диалога.

**Задача №6**

Написать программу, очищающую строку от лишних пробелов. Лишними счи-

таются пробелы в начале строки, в конце строки и пробелы между словами,

если их количество больше 1.

**Пояснение**

В данной программе запрещёно создавать дополнительные массивы, то есть необходимо стремиться к экономии памяти. Время выполнения программы значения не имеет.

**Состав**

Программа должна состоять из двух функций:

* char \* clear(char \* line) - очистка строки **line**.
* **main()** - организация диалога.

**Задача №7**

Написать программу, выводящую таблицу встречаемости символов для введен-

ной пользователем строки. В этой таблице содержится символ строки и число

его повторений.

**Пояснение**

В этой программе мы стремимся к экономии времени, так что использование дополнительных массивов оправдано

**Состав**

Программа должна состоять из функции:

* **main()** .